

Implémentation et Analyse d'une White-box du DES

David Wong Jacques Monin Hugo Bonnin

Université de Bordeaux

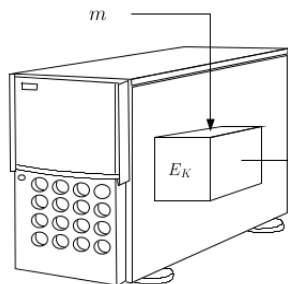
2014

A quoi ça sert ?

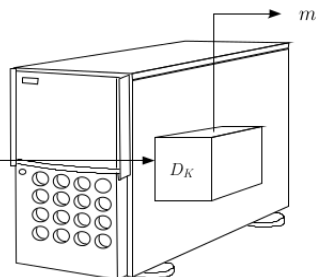


Base de la cryptographie

Alice



Bob



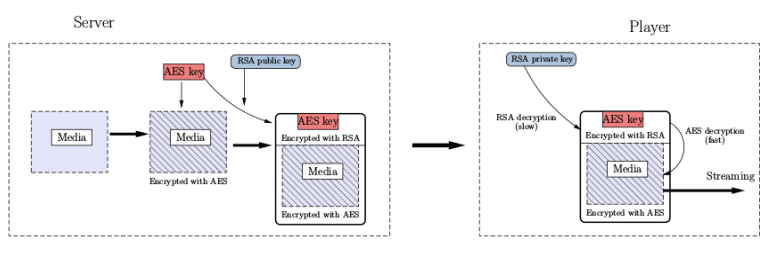
c
Eve

A horizontal line connects the E_K box of Alice's device to the D_K box of Bob's device. A zigzag arrow labeled c points upwards from the label "Eve" to this communication line.

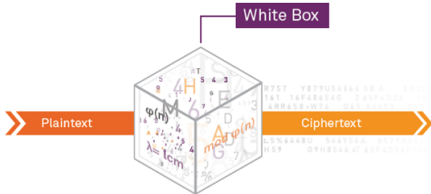
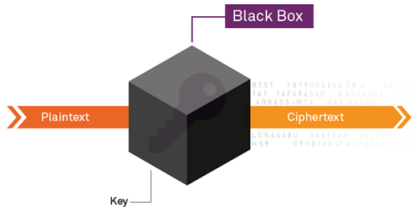
Man At The End

```
.-----.  
|      ATTAQUANT      |  
|  .-----  .  |  
|  |          |  |  
|  | PROGRAMME |  |  
|  |          |  |  
|  '-----'  |  
|              |  
| '-----'  |
```

Examples



Définition



Algorithme DES

- Le but est de transformer toutes ces opérations

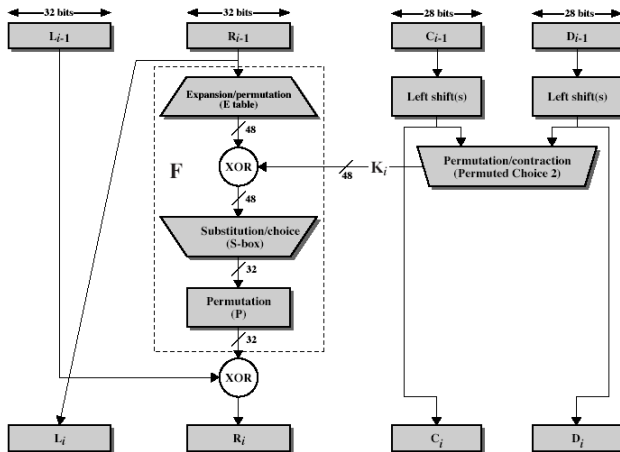







Figure 3.8 Single Round of DES Algorithm

- ▶ DES : www.github.com/mimoo/DES
- ▶ WHITEBOX-DES : www.github.com/mimoo/whiteboxDES

PUBLIC  mimoo / whiteboxDES

 Unwatch 4  Star 1  Fork 1

branch: master whiteboxDES / src / WBDES.c 

hbonnin 10 days ago decommenting / back to normal fonctionning

0 contributors

file 141 lines (117 slo) 2.799 kb Edit Raw Blame History Delete

```
1 #include "WBDES.h"
2 #include "tboxes.c"
3 #include <stdbool.h>
4 #include <stdio.h>
5 #include <stdint.h>
6 #include <stdlib.h>
7
8 // 64 bits input -> 96 bits output (M1)
9 void before_rounds(unsigned int *in, unsigned int *out)
10 {
11     unsigned int temp1[24][8];
12     unsigned int temp2[24];
13
14     // Fill temp2[24] with 0
15     for(int ii = 0; ii < 24; ii++)
16     {
17         temp2[ii] = 0;
18     }
19
20     // Sub-matrices
21     for(int ii = 0; ii < 24; ii++)
22     {
23         for(int jj = 0; jj < 8; jj++)
24         {
25             temp1[ii][jj] = M1_tables[ii][jj][in[jj]];
26         }
27     }
28
29     // Xor part
30     for(int ii = 0; ii < 24; ii++)
31     {
32         for(int jj = 0; jj < 8; jj++)
33         {
34             temp2[ii] = xorTables[(((temp2[ii] << 4) + temp1[ii][jj]))];
35         }
36     }
37 }
```


Partial evaluation

- ▶ Regrouper le XOR entre le bloc et la clé avec l'opération de substitution.
- ▶ On peut ensuite pré-calculer toutes les sorties possibles de cette opération.
- ▶ Les tables créées sont les seules du programme à être modifiées lorsqu'une nouvelle clé est utilisée.

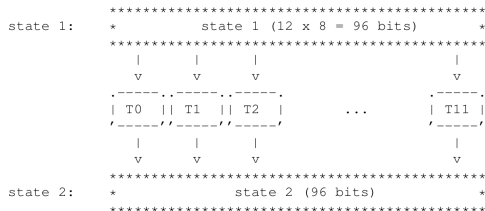
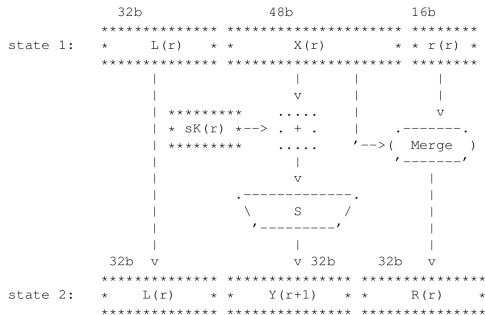
Tabularization

Entrée	S-box	Sortie																														
0010 := 2 -->	<table><tr><td colspan="6">.-----.</td></tr><tr><td> 0 </td><td> 1 </td><td> 2 </td><td> ... </td><td> 15 </td><td> </td></tr><tr><td colspan="6"> ----- </td></tr><tr><td> 5 </td><td> 2 </td><td> 0 </td><td> ... </td><td> 8 </td><td> </td></tr><tr><td colspan="6">,-----,</td></tr></table>	.-----.						0	1	2	...	15		-----						5	2	0	...	8		,-----,						--> 0 := 0000
.-----.																																
0	1	2	...	15																												

5	2	0	...	8																												
,-----,																																

FIGURE 1: Tabularisation

Transformation



Décomposition de Matrice

$$\begin{bmatrix} Y0 \\ \vdots \end{bmatrix} = \begin{bmatrix} M \end{bmatrix} \times \begin{bmatrix} X0 \\ \vdots \end{bmatrix}$$

$$\begin{bmatrix} Y0 \\ Y1 \end{bmatrix} = \begin{bmatrix} A & B & C & D \\ E & F & G & H \end{bmatrix} \times \begin{bmatrix} X0 \\ X1 \\ X2 \\ X3 \end{bmatrix}$$

FIGURE 2: Décomposition de Matrice

Input/Output Encoding

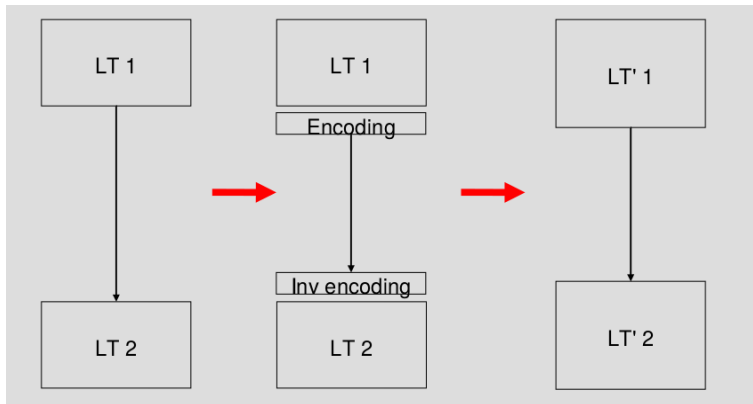


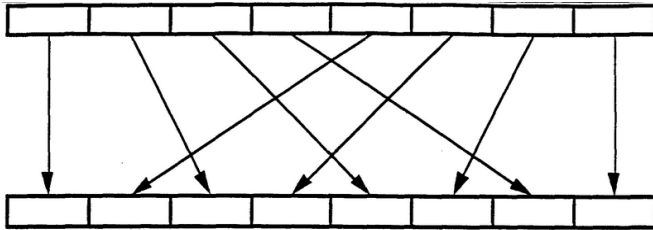
FIGURE 3: Encoding

Concepts secondaires

```
*****
*                state 2 (96 bits)                *
*****
|              |              |                               |
v              v              v                ...              v

????????????????????????????????????????????????????????
|              |              |                ...              |
v              v              v                               v
*****
*                state 3 (96 bits)                *
*****
```

Randomization

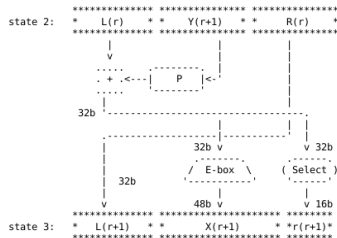


Mixing Bijection

[illegible]

$$G^{-1} \cdot (G \cdot M_1) \text{ où } G \cdot M_1$$

Bypass



- ▶ On empêche l'identification facile des opérations
- ▶ On rajoute des bits en entrée et en sortie

Combined Function

```
state 2:
```

```
***** L(r) ***** Y(r+1) ***** R(r) *****  
*****  
          |                               |                               |  
        v                               |                               |  
    ..... ,-----,                |                               |  
    . + .<---|      P      |<-'|   |                               |  
    ..... /-----'         |       |                               |  
    |                           |       |                               |  
32b -----|-----,           |       |                               |  
            |-----|-----'     |       |                               |  
            |               32b v       v 32b  
            |               / E-box \   ( Select )  
            | 32b          /-----'   /-----'  
            |               |               |  
            v              48b v             v 16b  
*****  
state 3:
```

```
***** L(r+1) ***** X(r+1) ***** *r(r+1)* *****  
*****
```

$$(P||Q)(input_P||input_Q).$$

Split-Path Encoding

Entrée

S-box

Sortie

0011||0010 --> .-----.
|...| 0011||0010 |...|
|-----|---|
|...| 0001 |...| --> 0001
,-----,

|
v

0011||0010 --> .-----.
|...| 0011||0010 |...|
|-----|---|
|...| 0001||xxxx |...| --> 0001||1001
,-----,

External Encoding

- ▶ Appliquer deux bijections à l'entrée et la sortie de DES
- ▶ $Whitebox = E \circ DES(input) \circ G$

Conclusion

- ▶ Beaucoup d'effort pour d'autres solutions (API, clés publiques)
- ▶ Taille importante
- ▶ La non-connaissance des algorithmes est "trop" importante.
- ▶ Utilisé professionnellement

